# SATへの変換を利用したグラフ問題の難しさの評価方法について(計算量理論)

AUTHOR(S):

宮崎, 修一; 岩間, 一雄

CITATION:

宮崎, 修一 ...[et al]. SATへの変換を利用したグラフ問題の難しさの評価方法について(計算量理論). 数理解析研究所講究録 1994, 871: 112-116

ISSUE DATE:

1994-05

URL:

http://hdl.handle.net/2433/84048

数理解析研究所講究録
第871巻 1994年 112-116


# SATへの変換を利用したグラフ問題の難しさの評価方法について

九州大学工学部　宮崎 修一　(Shuichi Miyazaki)
九州大学工学部　岩間 一雄　(Kazuo Iwama)

## 1　はじめに

NP完全問題は，問題のサイズに対して多項式時間で解くアルゴリズムが知られていない問題の代表例である．これらの問題は多項式程度の違いを無視すれば，同じ計算時間で解けるという点で，同じクラスに属する．即ち，ある問題を多項式時間で別の問題に変換することができる．これらの変換は，例えば問題のNP完全性を証明するのに用いられてきた．その場合，多項式時間であればどのような変換であってもかまわないという大雑把なものであったが，ある問題$A$を別の問題$B$に変換して，問題$B$に対する効率の良いアルゴリズムを使って解くという実用的応用を考えれば，変換の効率を良くすることが大切になる．

例えばCNF論理式の充足可能性問題（SAT）に対する局所探索法というアルゴリズムが最近発見された[1][3][4]．このアルゴリズムが1000変数程度の例題なら実用的時間で解けると仮定すると，$n$頂点のグラフ問題$A$を$n^2$変数の論理式に変換すれば30頂点の例題までしか解けないが，$n$変数の論理式に変換できれば1000頂点まで解くことができる．

そこで我々は，様々なNP完全問題をSATに変換するときに必要な変数の数に着目し，問題の難しさを議論することを考えた．つまり，変換の際に最低限必要となる変数の数が多いほど難しい問題であると考えるわけである[1][2]．NP完全問題は“手に負えない問題”として統一的に論じられることが多かったが，この方法を用いれば個々の問題の難しさの違いをかなりの具体性を持って議論できるのではないかと思われる．

また，この手法は発見的手法（ヒューリスティック）を評価するのにも役に立つと思われる．例えば，グラフの問題を解くときにグラフのある性質を利用するが，それがどれだけの効果を発揮するのかを数学的に解析するのは容易ではなく，これまでは実験的な評価が多かった．しかし，問題をSATに変換するときにその性質を用いればどれだけ変数を減らすことができるかを議論することによって，そのヒューリスティックの良さを評価できるであろう．

本論文では，$k$-クリーク問題をSATに変換する方法を考える．与えられるグラフの頂点数を$N$として，最初に$k\log N$変数での自然な変換方法を述べる．次に変数を減らし，$\log N+(k-1)\log D$変数（$D$は与えられたグラフ中の次数の大きい方から$k$番目の頂点の次数）および，$\frac{k}{3}\log m$変数（$m$はグラフ中の三角形の数）で変換できることを示す．以前我々は，$k$色の頂点彩色問題を自然な変換により，$N\log k$変数で変換できることを示した[2]．一般に，$k$-クリーク

問題はグラフの頂点彩色問題より易しいといわれているが，今回の結果はある意味でそれを裏付けたのではないかと思われる．

## 2 SAT および $k$- クリーク問題

SAT とは，与えられた CNF （和積形）論理式を 1 にする変数の割当があるかどうかを問う問題である．$k$- クリーク問題とは，$N$ 頂点のグラフと $k$ が与えられた時に，$k$ 頂点の完全部分グラフが含まれているかどうかを問う問題である．

## 3 変換アルゴリズム

### 3.1 自然な変換

まず，頂点数 $N$ の $k$- クリーク問題を $k \log N$ 変数で SAT に変換する方法を述べる．使用する変数は，$x_{i,j}(1 \leq i \leq k, 1 \leq j \leq \log N)$ で，その使い方は，$(x_{i,1}, x_{i,2}, \cdots, x_{i,\log N})$ に入る 0,1 の 2 進列で，クリークに含まれる頂点の番号を表すことにする．ただし，$x_{i,1}$ が低いビットを表すことにする．例えば，$(x_{2,1}, x_{2,2}, x_{2,3}, x_{2,4}, \cdots, x_{2,\log N})$=$(1,0,1,0,\cdots,0)$ というのは，頂点 5 がクリークに含まれる頂点として選ばれたことを意味する．この方法で，$(x_{i,1}, x_{i,2}, \cdots, x_{i,\log N})(1 \leq i \leq k)$ に重複なく頂点を選び，しかもその $k$ 個の頂点のうちどの 2 つをとっても間に枝があるとき，作られた論理式は充足可能である．頂点の番号は 0 ～ $N-1$ で表すことにする．

[ 変換アルゴリズム **clique_to_SAT** ]

**ステップ 1：** $1 \leq i \leq k$ の各 $i$ に対して $(x_{i,1}, x_{i,2}, \cdots, x_{i,\log N})$ が 2 進の $N-1$ を超えると 0 となる節をつくる．

**ステップ 2：** $1 \leq i < j \leq k$ の各 $i, j$ に対して，$(x_{i,1}, x_{i,2}, \cdots, x_{i,\log N})$ の表す値と $(x_{j,1}, x_{j,2}, \cdots, x_{j,\log N})$ の表す値が同じとき 0 となる節をつくる．

**ステップ 3：** 間に枝のない頂点を $a$,$b$ とするとき，$1 \leq i < j \leq l$, の各 $i, j$ に対して，次の 2 つの節をつくる．

$(x_{i,1}, x_{i,2}, \cdots, x_{i,\log N})$ が $a$ を，$(x_{j,1}, x_{j,2}, \cdots, x_{j,\log N})$ が $b$ を表したとき 0 となる節と

$(x_{i,1}, x_{i,2}, \cdots, x_{i,\log N})$ が $b$ を，$(x_{j,1}, x_{j,2}, \cdots, x_{j,\log N})$ が $a$ を表したとき 0 となる節．

この操作は，すべての $a$,$b$ の組合せに対して行なう．

ステップ 1 は，$k$ 個選ばれる頂点の番号以外を表してはいけないという条件を式にしたものである．例えば，$x_{i,1}, x_{i,2}, \cdots, x_{i,5}$ が 25 を 2 進数で表した値 $(1,0,0,1,1)$ を超えないようにするためには，$(\overline{x_{i,2}} + \overline{x_{i,4}} + \overline{x_{i,5}})(\overline{x_{i,3}} + \overline{x_{i,4}} + \overline{x_{i,5}})$ という節をつくれば良い．ステップ 2 では同じ頂点が重複して選ばれないようにしている．つまり，頂点の番号を表す 0 から $N-1$ の同じ値を

表してはいけないので，各$i,j$の組合せに対して$N$個の節をつくるわけである．例えば，共に5を表してはいけないという節は$(\overline{x_{i,1}}+x_{i,2}+\overline{x_{i,3}}+x_{i,4}+\cdots+x_{i,\log N}+\overline{x_{j,1}}+x_{j,2}+\overline{x_{j,3}}+x_{j,4}+\cdots+x_{j,\log N})$という節をつくる．またステップ3では，間に枝のない頂点のペアが選ばれないようにしている．

## 3.2 clique_to_SAT II

ここでは，$\log N+(k-1)\log D$($D$は与えられたグラフ中の次数の大きい方から$k$番目の頂点の次数）変数での変換を示す．前節では，頂点を表すのに頂点番号をそのまま使っていたが，それを工夫することにより変数を節約できる．$k$個選ばれる頂点の中で，1つを代表と考え，それはこれまで通り頂点番号で表すことにし，残り$k-1$個の頂点は，代表となる頂点の何番目の枝につながっているかで表すことにする．したがって，$k-1$頂点分は枝の数を表す長さのビット数が必要になるが，選ばれた$k$個の頂点のうちいちばん次数の小さいものを代表となる頂点にすればよいので，枝を表す変数は最悪の場合でも$k$番目に大きい次数を表せれば良い．

変数は$x_{1,j},(1\leq j\leq \log N),y_{i,j},(2\leq i\leq k,1\leq j\leq \log D)$を用い，$x$を代表となる頂点の番号を表すのに使い，$y$をその頂点から出る枝番号を表すのに用いることにする．例えば，$(x_{1,1},x_{1,2},x_{1,3},x_{1,4},\cdots,x_{1,\log N})$=$(1,0,1,0,\cdots,0)$，$(y_{5,1},y_{5,2},y_{5,3})=(1,1,0)$というのは，頂点5と，頂点5の3番目の枝につながっている頂点がクリークの頂点として選ばれたことを意味する．各頂点に対しては，そこから出ている枝に番号を0からその頂点の次数分までつけておくことにする．

[ 変換アルゴリズム **clique_to_SAT II** ]

**ステップ1：** $x_{1,1},x_{1,2},\cdots,x_{1,\log N}$が2進の$N-1$を超えると0となる節をつくる．

**ステップ2：** 次数$k-2$以下の頂点を$a$とする．各$a$に対して，$x_{1,1},x_{1,2},\cdots,x_{1,\log N}$が2進で$a$を表したら0となる節をつくる．

**ステップ3：** $1\leq i\leq N$,$2\leq j\leq k$の各$i$,$j$に対して$x_{1,1},x_{1,2},\cdots,x_{1,\log N}$が$i$を表し，かつ$y_{j,1},y_{j,2},\cdots,y_{j,\log D}$が（頂点$i$の次数$-1$）を超えたとき0となる節をつくる．

**ステップ4：** $2\leq i<j\leq k$の各$i,j$に対して，$y_{i,1},y_{i,2},\cdots,y_{i,\log D}$の表す値と$y_{j,1},y_{j,2},\cdots,y_{j,\log D}$の表す値が同じとき0となる節をつくる．

**ステップ5：** $1\leq v\leq N$の各$v$に対して，以下のようにする．

頂点$v$とつながっている頂点のうち互いに枝のないものを$a$,$b$とする．また，頂点$v$の枝で頂点$a$,$b$につながる枝をそれぞれ$e_a,e_b$とする．$2\leq i<j\leq k$の各$i,j$に対して，以下の2つの節をつくる．

$x_{1,1},x_{1,2},\cdots,x_{1,\log N}$が$v$を表し，$y_{i,1},y_{i,2},\cdots,y_{i,\log D}$が$e_a$を表し，$y_{j,1},y_{j,2},\cdots,y_{j,\log D}$が$e_b$を表したとき0となる節と，

$x_{1,1}, x_{1,2}, \cdots, x_{1,\log N}$ が $v$ を表し，$y_{i,1}, y_{i,2}, \cdots, y_{i,\log D}$ が $e_b$ を表し，$y_{j,1}, y_{j,2}, \cdots, y_{j,\log D}$ が $e_a$ を表したとき 0 となる節.

次数が $k-2$ 以下の頂点は $k$ 頂点の完全グラフを形成しない．ステップ 2 でそのような頂点が選ばれたら 0 となる節をつくっている.

グラフ $G$ の頂点間に枝が存在する確率を $p$ とする．$D$ はグラフ $G$ 中で $k$ 番目に大きい次数であるが，$G$ の平均次数を $D$ と考えると，$D \simeq pN$ となり，$k\log N - (k-1)\log\frac{1}{p}$ 変数での変換が可能である.

### 3.3 clique_to_SAT III

次に，$k = 3l$ の場合（一般の場合へ拡張可）を考える．まず，与えられたグラフから 3 頂点完全グラフ（三角形）を全て取り出し，0 から $m-1$ まで番号をつける．そして，2 つの三角形が頂点を共有しないとき互いに独立であるといい，2 つの三角形間の全ての頂点間に枝があるとき互いに隣接するということにする．このとき，$k$-クリークは，独立であり，かつ隣接する三角形が $l = \frac{k}{3}$ 個集まったものと見ることができる．クリークとして選ばれる $l$ 個の三角形の番号を表すために $l\log m$ 個の変数が必要である．変数は $x_{i,j}, (1 \le i \le l, 1 \le j \le \log m)$ を用いる.

[ 変換アルゴリズム **clique_to_SAT III** ]

**ステップ 1：** $(x_{i,1}, x_{i,2}, \cdots, x_{i,\log m})$ が 2 進の $m-1$ を超えると 0 となる節をつくる.

**ステップ 2：** $1 \le i < j \le l$ の各 $i, j$ に対して，$(x_{i,1}, x_{i,2}, \cdots, x_{i,\log m})$ の表す値と $(x_{j,1}, x_{j,2}, \cdots, x_{j,\log m})$ の表す値が同じとき 0 となる節をつくる.

**ステップ 3：** 間に枝のない頂点を $a$,$b$ とするとき，$1 \le i < j \le l, i \ne j$ の各 $i, j$ に対して，以下の 2 つの節をつくる.

$(x_{i,1}, x_{i,2}, \cdots, x_{i,\log N})$ が $a$ を，$(x_{j,1}, x_{j,2}, \cdots, x_{j,\log N})$ が $b$ を表したとき 0 となる節と，

$(x_{i,1}, x_{i,2}, \cdots, x_{i,\log N})$ が $b$ を，$(x_{j,1}, x_{j,2}, \cdots, x_{j,\log N})$ が $a$ を表したとき 0 となる節.

この操作は，すべての $a$,$b$ の組合せに対して行なう.

グラフ $G$ から 3 頂点を選ぶ組合せは ${}_NC_3 \simeq \frac{n^3}{6}$ であり，任意に選んだ 3 頂点が三角形をなす確率は $p^3$ であるので，$m \simeq \frac{1}{6}(pN)^3$ である．この方法では $\frac{k}{3}\log m \simeq k\log N - k\log\frac{1}{p}$ 変数での変換が可能となる.

### 3.4 clique_to_SAT IV

**clique_to_SAT III** の方法は $m$ 個の三角形を頂点とし，独立であり，かつ隣接する 2 つの三角形に対応する頂点間に枝があるグラフとみなせば $m$ 頂点グラフ $G'$ の $\frac{k}{3}$-クリーク問題と考えられるため，更に **clique_to_SAT II** の方法を適用することにより変数を減らすことができ

る．この場合は，$G'$ 中の $\frac{k}{3}$ 番めに大きい次数を $D'$ として $\log m + (\frac{k}{3}-1)\log D'$ 変数での変換が可能である．

$N$ 頂点からできる三角形の数は $m = p^3 \cdot_N C_3$ で，ある1つの三角形を選んだときそれと独立である三角形は $m' = p^3 \cdot_{N-3} C_3$ なので，独立である確率は $\frac{m'}{m} \simeq 1$ である．また，任意の2つの三角形が隣接である確率は $p^9$ なので，グラフ $G'$ の2頂点間に枝が存在する確率は $p^9$ となる．よって必要な変数の数は $\log m + (\frac{k}{3}-1)\log p^9 m = k\log pN + 3(k-3)\log p$ となる．ここで注意すべき点は，$p$ の値が $N^{-\frac{1}{4}}$ より小さい場合はこの値が負になってしまうことである．これは，グラフ $G'$ の平均次数が $mp^9 = N^3 p^{12}$ であり，$p < N^{-\frac{1}{4}}$ の場合この値は1以下になるためである．即ちこの場合グラフ $G$ 中には独立かつ隣接な三角形は存在しないことになる．

### 3.5 考察

**clique_to_SAT II** と **clique_to_SAT III** はグラフの異なる性質を利用したものであるが，グラフ $G$ の平均次数を $D$ とすれば，任意の頂点間に枝が存在する確率は $D/N$ であり，任意の3頂点を選んだときそれが三角形を構成する確率は $(D/N)^3$ である．3頂点の選び方は ${}_NC_3 \simeq N^3/6$ 通りあるので，三角形は平均 $D^3/6$ 個あることになる．よって，**clique_to_SAT III** の方法では $\frac{k}{3}\log m \simeq k\log D$ 変数必要であることになり，**clique_to_SAT II** の方法とほとんど同じ数の変数が必要になる．

## 4 おわりに

本稿では，$k$-クリーク問題をSATに変換するときの変数の数について議論してきたが，必要な変数の下限を求めることが，まだできていない．今後我々は，様々なアプローチから下限を示す研究を進めるとともに，下限は存在するのかどうか，また，オーダー的になら下限を求めることができないか，という観点からも研究を進めていきたい．

## 参考文献

[1] 宮崎，岩間．CNF論理式に対する局所探索法の評価．平成4年度情報処理学会アルゴリズム研究会（平成5年3月）．

[2] 宮崎，岩間．グラフの色ぬり分け問題からSATへの効率の良い変換方法について．平成5年夏のLAシンポジウム（平成5年7月）．

[3] E. Koutsoupias and C. Papadimitriou. On the greedy algorithm for satisfiability. *Information Processing Letters 43*, pp.53-55,1992.

[4] D. Mitchell, B. Selman and H. Levesque. Hard and easy distributions of SAT problems. In *Proc. Tenth National Conference on Artificial Intellegence*. pp.459-465,1992.